Interpolated Rigid Map Neural Networks for Anatomical Joint Constraint Modelling

Glenn Jenkins, Paul Angel

Cardiff School of Technologies
Cardiff Metropolitan University
Cardiff, South Wales, UK.
Email: gljenkins@cardiffmet.ac.uk, pnangel@cardiffmet.ac.uk

***Abstract* - The demand for accurate individual and general kinematic joint models is increasing with growing applications in fields such as animation, biomechanics, motion capture, ergonomics and robot human interaction modelling. Many approaches have exploited unit quaternions to eliminate singularities when modelling orientations between limbs at a joint, leading to the development of a number of novel joint constraint validation and correction methods. A number of machine learning approaches have been applied to this modelling problem, as depending on training data either individual or general joint models can be created. Recent work has demonstrated the use of Rigid Maps to model regular conical constraints on the orientation of the limb. In this paper we extend this work deploying a modified Rigid Map Network with a continuous output.**

Keywords - Rigid Map; Unit Quaternion; Joint Constraint; Neural Network

I. INTRODUCTION

Joint constraint systems are essential components of anatomical models, ensuring the correct movement of limbs during simulation. Such models are required in a number of fields, in animation they can be used to produce realistic articulated figures [1]. In computer vision applications anatomical models (or priors) are fitted to recovered image or motion capture data [2]. Anatomical models are also used in biomechanical simulations [3] to model patients and in ergonomics [4] modelling average individuals. They are also required in simulations towards the development of control systems for personal assistance robots [5].

Dynamics-based models are increasingly used in biomechanics and rehabilitation research in bimodal approaches combining biomechanical models with experimental data (e.g. measured external forces) [6]. Dynamics solutions can be used to produce realistic joint behavior based on input, contact and constraint forces [7]. The in vivo measurement of model parameters for example joint torques, contact forces and constraint forces from muscles etc. via non-invasive techniques are problematic and the verification of estimated values both time consuming and error-prone [6]. Where accurate biological simulation is not required, kinematic joint limits can be enforced using constraint forces generated from kinematic data [5]. Inverse-Kinematics (IK) based approaches allow the precise placement of end effectors as constraints [8]. IK solvers attempt to resolve a system of constraints, but problems arise due to the existence of multiple potentially suboptimal solutions [8].Well known applications of IK to model human limbs include interactive character control [9], path planning [10], motion retargeting [11], [12] and Motion Capture clean up [13]. IK solvers can be classified as: analytical, often resorting to reduced coordinate formalisms; numerical, using iterative approaches to solve a system of constraints, or a combination of these [11], [14]. Additional constraints in the form of pose constraints (valid poses,) or rotational joint limits can be used to reduce the number of potential solutions, an important aspect of this is how joint constraints are represented [15], [16].

This work builds on previous research exploring the use of unsupervised neural networks to model unit quaternion based phenomenological [17] joints (whose behavior is modelled without reference to the underlying joint anatomy). Rigid Maps [18] (derived from Kohonen's SOM), can be used to implicitly model valid orientations (expressed as unit quaternions) [19]. In this paper constraints on the rotation of the limb (or swing [20]) with regular (circular) bounded constrained regions are considered. Irregular boundaries and rotation around the limb (or twist [20]) are the subject of future work. This is a continuation of our earlier work [19], [21] exploring augmentations to the Rigid Map to produce a continuous rather than a discrete output.

II. RELATED WORK

Joint constraints can be expressed using Euler angles but the resulting models can encounter singularities [22]. Kinematic models (known as priors) with box-limit joint constraints representing valid poses have been in the estimation of pose from recorded images [15] and the animation generation from recorded samples [16]. Analytical methods for rotational constraint boundaries have also been explored [23], [24]. Geometric constraints can also be used to approximate these boundaries including spherical [25], [26] and conical polygons [27].

Engell-Nørreg°ard et al [28], [29] proposed signed distance cones. In this approach a lattice with cubic or cuboidal cells represents rotation at a given resolution along each axis. The valid region is sampled, and cells are marked as being interior or exterior. Aklher and Black [30] use a similar two dimensional discretization (described as an occupancy matrix,) here polar coordinates describe the orientation of the shoulder, hip and neck relative to the torso. Here a hemispherical half space is defined by a plane at the joint and a box region fitted to points which result from the projection of the limb end point on the plain.

Unit quaternion algebra allows the representation of rotation without the presence of Gimbal Lock [22]. Quaternions are hyper-complex numbers, composed of one real and three imaginary components where $q = \langle s, i, j, k \rangle$. Multiplying complex numbers results in rotation in the complex plane, giving rise to $i^2 = -1$. This is extended in quaternion space, where $i^2 = j^2 = k^2 = -1$. Unit quaternions occupy a three dimensional surface (an S^3 hypersphere) in four dimensional space ($\mathbf{R}^4$) and can be used to represent rotations. This representation is redundant as the unit quaternions represent 4π rotations, polar opposites (q and $-q$) represent the same orientation [22] i.e. they are antipodal.

Several joint constraint methods have used this parametrization, the simplest decomposed the unit quaternion into conical and axial components (also unit quaternions,) constrained independently [31]. This was extended to model the relation between the conical and axial constraint components based on recorded samples [32].

A number of approaches leverage volumetric modelling of a sampled data a set of unit quaternions representing valid orientations projected as a point cloud in three-dimensions. This removes the $q = -q$ ambiguity in quaternion space and allows valid orientations to be represented as a 3D volume with a boundary. Both spherical primitives [33] and voxels [33] have been used to describe these regions. An iterative approach was employed in both to resolve invalid joint configurations, rotating toward the nearest primitive (sphere or voxel,) until the given orientation was valid.

Statistical approaches have also been employed, Johnson [34] projected one half of the unit quaternion hypersphere into a three dimensional tangent space, allowing pose and joint constraints to be implemented in terms of a maximum deviation from the mean of the projected points [34]. Iterative correction towards the mean, to within the constraint and reverse projection can be used to correct an invalid orientation [34]. Recently Brau and Jiang [35] used a four dimensional Von Mises-Fisher distribution to describe unit quaternion over a hypersphere in R4. Its probability density function is parametrized by mean direction and concentration parameters, for which maximum likelihood estimates are obtained per-joint from a motion capture database [35].

Artificial Neural Networks (ANNs) are inspired by the structure and mechanics of biological neural networks but differ significantly in their complexity and inter-neuron communications [36].

Jiang and Liu [5] used a fully connected feed-forward neural network and supervised training to model a hypersurface representing the range of motion of a number of collected limbs. This produced a differentiable function whose gradient could be used in robotics applications (for example creating a dynamic constraint to enforce physical joint limits). Generalized Multi-layer Perceptron (or GMLP) Neural Networks with an evolved topology have been used to implicitly model a joint constraint boundary [37]. Errors in the neural network approximation results in some correction of valid orientations along with over and under correction of invalid orientations. The former may be partly overcome using a classifier to identify orientations requiring correction (for example an SVM [38]). These supervised training methods require training data which includes valid and invalid input and output, this makes them difficult to apply to recorded data (for example from motion capture).

Jiang and Liu [5] overcome this by generating unlimited training data. Here joint configurations are uniformly sampled and labelled using the validation method proposed by Akler and Black [30]. An alternative is to make use of ANNs trained using unsupervised techniques such as competitive learning.

Self-Organizing Maps (SOMs) are trained using competitive learning and composed of two layers [39]. SOMs have been trained to implicitly model a simplified joint constraint using only valid orientations expressed as unit quaternions [40]. The output nodes of the network are trained via competitive learning to represent the training data while preserving the topography of the input space. The network responds to a given input orientation with the closest orientation in its model of the input data. This can be used directly for correction [40], or with an iterative approach [41] like that employed by Herda et al [33]. SOMs have also been extended into the domain of complex numbers with weights and distance metrics in unit quaternion space [41].

The Rigid Map Network is a modified SOM proposed for pose estimation problems by Winkler et al [18] and applied to joint constraint modelling in our earlier work [19], [21]. Here self-organization is no longer required, each output node represents a position (pi) in a known orientation space. The output node topology is fixed, and the nodes are uniformly distributed over the orientation space, a S3 hypersphere, using regular polyhedral or other techniques for uniform distribution.

The learning algorithm is modified so that the winning node (w) is based the proximity between the input pattern ($\hat{\boldsymbol{w}}$) and the position (p_i) of the output node rather than its weight determined by the inner product. The updating of weights, however remains unchanged with the winning node and its neighbors being updated; when fired the

network responds with the weight which is the shortest Euclidean distance from the input [42]. For a more detailed description of the learning algorithm please see [19]. Regular polytopes were initially used to uniformly distribute the output nodes in unit quaternion space (the surface of an S^3 hypersphere) [19]. Increasing the output layer density using the OSPHERE algorithm proposed by Fishman [43] improved these results [21].

Winkler [42] explored the use of interpolation techniques previously applied to SOMs [44], to produce a continuous output. The weight of the winner $(\hat{w})_w$ along with n runner-up neurons w_1 , ... , w_n are selected and combined, approaches include distance based methods where the output of the winner and n runner up nodes are combined using an inverse weighted sum [44] or an exponential function [45]. These approaches work on the assumption that a better output can be found between the n closest nodes and that neighborhoods in the input and output space are similar, to overcome these limitations geometrical and topological interpolation techniques have been suggested [44].

Geometric interpolation [44] makes use of both distance and direction information from the input space. The remaining error (from winner's weight to input) is projected in the direction of the next runner-up. The scalar product is used to calculate the optimal fraction required in the direction of the next runner up. Topological interpolation [44] this uses the same method for combining weights, but neighboring nodes in the output layer topology are selected rather than those with similar weights.

It is hypothesized that the Rigid Map Network with interpolation will produce superior results to the earlier Rigid Map. As in our previous work the focus is a regular rotational boundary with zero rotation around the limb.

The remainder of this paper is structured as follows: Section 3 provides a description of our methodology with reference to the techniques employed. Section 4 reports the results of the experiments undertaken with these discussed in Section 5. Finally Section 6 draws conclusions from this work and highlights areas for future investigation.

III. METHODOLOGY

This paper describes an extension to our earlier work modelling valid orientations of a virtual limb parametrized using unit quaternions using a Rigid Map neural network. The network consists of four input nodes each with a weighted connection to a number of output nodes. The output nodes are placed into a topology each having a position in S3. These were initially arranged using both regular polytopes in 4D-space and a number of uniformly distributed points over S3 [21].

Rigid Maps were trained to identify the closest valid orientation for both valid and invalid input, here the input layer represents the current virtual limb orientation as a unit quaternion. The weighted connections of the winning output node (or interpolated winning nodes) represent the nearest valid orientation. The performance of the augmented Rigid Map in the context of anatomical constraint modelling was explored using regular constrained regions of varying sizes.

TABLE I. TRAINING CONFIGURATION

Parameter	*Description*	*Setting*
Input nodes	Number of input nodes	4
Training patterns	Number of training patterns.	1000
Initial learning rate (λ)	Initial value ($\lambda(0)$), this decays exponentially at each for time step t with constant k.	1.0
Learning rate decay constant(k_λ)	Constant used to control rate of learning rate decay.	0.15
Initial neighbourhood radius (σ)	Initial value ($\sigma(0)$), this decays exponentially at each for time step t.	2π Radians
Neighbourhood rate decay constant (k_σ)	Constant used to control rate of learning rate decay.	1.0
Neighbourhood constant (ϕ)	Constant used control the neighbourhood function.	0.04
Maximum training epochs	Maximum number of time steps.	1000
Distance metrics	Euclidean Distance Direction Cosine	$E = (\|(q^a - q^b)\|)^2$ $D = arccos(q^a \cdot q^b)^2$

The training process is detailed in our earlier work [21] for each training epoch the input set is presented to the network. A squared distance (D) is calculated between each input pattern and the topological position of the output node is calculated using a given metric (detailed in TABLE I). The weight of node with the shortest distance (the winning node) is moved towards the input according to the learning rate. Output nodes within the neighborhood (topological regions according to position) of the winning node also have their weights updated. The learning rate and neighborhood decrease exponentially with time. The constants used are provided in TABLE I along with the ranges used for experimentation. Each Rigid Map was trained until it converged, or a maximum number of training epochs was reached.

The usage phase begins with an input pattern being presented to the trained Rigid Map. The network responds with the weight of the winning node using a second distance metric (E) between the input and the weight (the connections between the input nodes and a given output node). At this stage various interpolation techniques were applied including a inverse weighted average of the n best nodes along with two variations of geometric interpolation [44]. Each experiment was repeated ten times to ensure the consistency of the results.

Experiments were undertaken with output layers containing 720 nodes distributed using combination of a polyFixed Point Invalid Region Valid Region Regular Boundary tetrahedron and poly-dodecahedron along with a uniform distribution containing 2000 nodes.

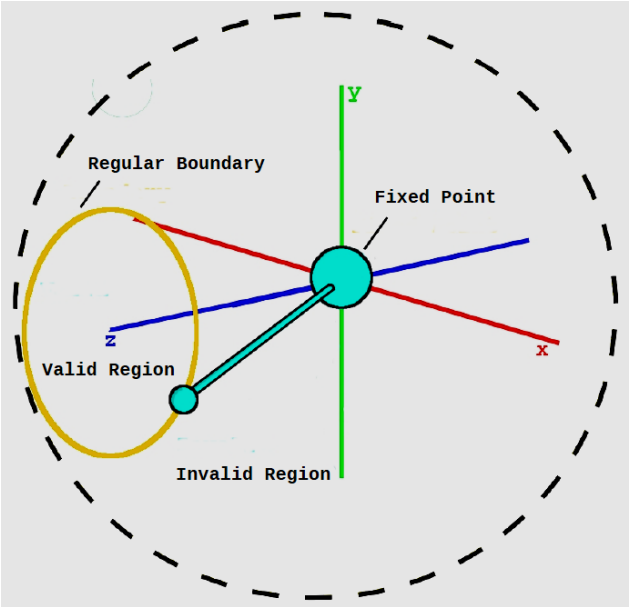

Figure 1. Model used for dataset generation. Valid region inside boundary invalid region outside.

Training and testing datasets contained 1000 patterns. In experiments where the range was not varied, a constant range of 20 was used. Defaults for the other parameters are given in TABLE I and were identified though experimentation. The training dataset contained only valid patterns, like those recorded from the movement of a human arm (see Figure 1). A set of 'ideal' corrected orientations were generated using Lee's [31] approach and provided a measurement of the Rigid Maps capabilities (see [21] for details).

IV. RESULTS

The results show the effect of correcting the orientation to that suggested by the Rigid Map and indicate successful training. An increase in the range of the constrained region results in a decrease in performance as shown in Figures 2(a) and 2(c). The results for the polytope distribution (with 720 output nodes) shown in Figure 2(a) demonstrates a small improvement over the discrete case. The much larger uniformly distributed output layer (with 2000 nodes,) produces much larger improvement (shown in Figure 2(b)) in the case of geometric interpolation. The number of nodes used for interpolation were selected based on the results shown in 2(c) and 2(d).

For polytope distributed output layer (720 nodes) a smaller number of interpolation nodes produced a better result (as shown in 2(c)). This result is reversed for the larger uniformly distributed output layer shown in Figure 2(d). In all cases the geometric interpolation approach produces superior results.

V. DISCUSSION

Rigid Maps are capable of modelling region occupied by valid orientations in unit quaternions space which can be used as a target for correction. The Mean Squared Error (MSE) on the test set (containing invalid and valid orientations) is comparable to those from our earlier SOM based approach [40]. Previous research [21] highlighted an increase in over correction of valid orientations as the valid region grows as output nodes are more dispersed over the valid region along with an increase in over correction of invalid orientation as fewer output nodes occupy spaces near the boundary. This can be reduced by increasing the density of the output layer [21].

The results presented show that interpolating between the winner and runner-up nodes produces a further reduction in this error (shown in Figure 2(a) and 2(b)). Where ideal orientation for correction lies between the winner and runner up nodes either within the valid region or near the boundary, then interpolating between these produces a more accurate result. Geometric interpolation, the later produced superior results (shown in 2) this suggests that both the contribution of the winning node and vector of interpolation are significant. A greater improvement was observed at a higher output node density of (as shown in 2(b)), this is accompanied by an improvement in results with a larger number of interpolating nodes (as shown in 2(d)). This suggests that with a sparse output layer the runner up nodes selected for interpolation not sufficiently similar and the effects of interpolation are negative. When the output layer is dense more nodes are required in the geometric approach to ensure that there are sufficient nodes to balance any movement of the winning node.

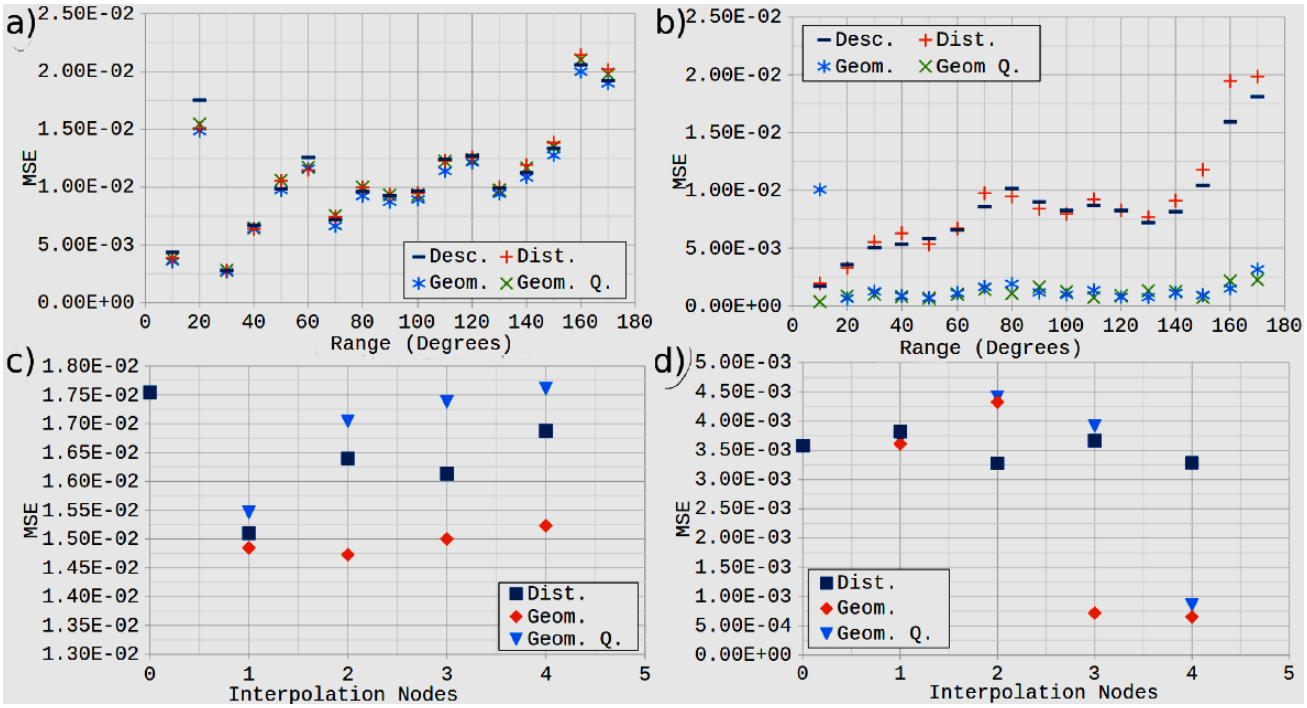

Figure 2. Graph (a) shows the range of the constraint for discrete and selected interpolation methods, with 720 output nodes distributed using a combination of polytopes. Graph (b) as (a) with 2000 output nodes distributed uniformly. Graph (c) shows the effect of increasing the number of nodes used for interpolation (zero is discrete), with 720 output nodes. Graph (d) as (c) with 2000 uniformly distributed output nodes.

VI. CONCLUSION

The augmentation of the Rigid Map with interpolation produces some improvement provided the output layer is suffciently dense and the number of interpolating nodes is high. Further experimentation with the interpolated Rigid Map is required to quantify the improvement of the interpolated result over the discrete case for valid and invalid orientations. Interpolation can also be applied to SOMs along with other networks which adjust their topological relations during unsupervised learning in-line with their training this an avenue of future research. Comparisons with other approaches [26], [28], [29], [40] in terms of accuracy and efficiency are now required.

Using a similar approach to Jiang and Liu [5] motion capture training and test sets could be developed for the Rigid Map, either for a single joint or multiple inter-related joints.

Current results are encouraging and suggest that Rigid Maps can implicitly model constraints on the rotation of the limb with regular boundaries in unit quaternion space. Their potential for modelling similar constraints with irregular Figure 2. Graph (a) shows the range of the constraint for discrete and selected interpolation methods, with 720 output nodes distributed using a combination of polytopes. Graph (b) as (a) with 2000 output nodes distributed uniformly. Graph (c) shows the effect of increasing the number of nodes used for interpolation (zero is discrete), with 720 output nodes. Graph (d) as (c) with 2000 uniformly distributed output nodes. Boundaries and rotation around the limb must now be explored.

REFERENCES

[1] L. Zhu, X. Hu, and L. Kavan, “Adaptable anatomical models for realistic bone motion reconstruction,” in Computer Graphics Forum, vol. 34, no. 2. Wiley Online Library, 2015, pp. 459–471.

[2] J. Chen, S. Nie, and Q. Ji, “Data-free prior model for upper body pose estimation and tracking,” IEEE Transactions on Image Processing, vol. 22, no. 12, pp. 4627–4639, Dec 2013.

[3] R. Riemer, E. T. Hsiao-Wecksler, and X. Zhang, “Uncertainties in inverse dynamics solutions: a comprehensive analysis and an application to gait,” Gait & posture, vol. 27, no. 4, pp. 578–588, 2008.

[4] X. Wang, “A behavior-based inverse kinematics algorithm to predict arm prehension postures for computer-aided ergonomic evaluation,” J Biomech, vol. 32, no. 5, pp. 453 – 460, 1999. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021929099000238

[5] Y. Jiang and C. K. Liu, “Data-driven approach to simulating realistic human joint constraints,” in IEEE International Conference on Robotics and Automation (ICRA), vol. abs/1709.08685, May 2018.

[6] A. H. Vette, T. Yoshida, T. A. Thrasher, K. Masani, and M. R. Popovic, “A comprehensive three-dimensional dynamic model of the human head and trunk for estimating lumbar and cervical joint torques and forces from upper body kinematics,” Medical engineering & physics, vol. 34, no. 5, pp. 640–649, 2012.

[7] P. M. Issacs and F. C. Micheal, “Controlling dynamic simulation with kinematic constraints, behaviour functions and inverse dynamics,” ACM Trans Graph, vol. 21, no. 4, pp. 215–223, 1987.

[8] A. Watt and M. Watt, Forward vs Inverse Kinematics in Computer Animation. New York: ACM Press, 1992, pp. 371–384.

[9] M. Kallmann, "Analytical inverse kinematics with body posture control," Computer Animation and Virtual Worlds, vol. 19, no. 2, pp. 79–91, 2008.

[10] D. Berenson, S. S. Srinivasa, and J. Kuffner, "Task space regions: A framework for pose-constrained manipulation planning," The International Journal of Robotics Research, p. 0278364910396389, 2011.

[11] H. J. Shin, J. Lee, S. Y. Shin, and M. Gleicher, "Computer puppetry: An importance-based approach," ACM Trans. Graph., vol. 20, no. 2, pp. 67–94, Apr. 2001. [Online]. Available: http://doi.acm.org/10.1145/502122.502123

[12] M. Meredith and S. Maddock, "Adapting motion capture data using weighted real-time inverse kinematics," Computers in Entertainment (CIE), vol. 3, no. 1, pp. 5–5, 2005.

[13] L. Kovar, J. Schreiner, and M. Gleicher, "Footskate cleanup for motion capture editing," in Proceedings of the 2002 ACM SIGGRAPH/ Eurographics symposium on Computer animation. ACM, 2002, pp. 97–104.

[14] J. Lee and S. Y. Shin, "A hierarchical approach to interactive motion editing for human-like figures," in Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, ser. SIGGRAPH '99. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1999, pp. 39–48. [Online]. Available: http://dx.doi.org/10.1145/311535.311539

[15] C. Sminchisescu and B. Triggs, "Estimating articulated human motion with covariance scaled sampling," The International Journal of Robotics Research, vol. 22, no. 6, pp. 371–391, 2003. [Online]. Available: https://doi.org/10.1177/0278364903022006003

[16] J. Min, Y.-L. Chen, and J. Chai, "Interactive generation of human animation with deformable motion models," ACM Trans. Graph, vol. 29, no. 1, pp. 9:1–9:12, Dec. 2009. [Online]. Available: http://doi.acm.org/10.1145/1640443.1640452

[17] A. E. Engin and S. T. Tumer, "Improvised dynamic model of human knee joint and its response to loading," J. Biomech. Eng., vol. 115, no. 2, pp. 137–142, 1993.

[18] S. Winkler, P. Wunsch, and G. Hirzinger, "A feature map approach to pose estimation based on quaternions," in Artificial Neural Networks ICANN'97, ser. Lecture Notes in Computer Science, W. Gerstner, A. Germond, M. Hasler, and J.-D. Nicoud, Eds. Springer Berlin Heidelberg, 1997, vol. 1327, pp. 949–954. [Online]. Available: http://dx.doi.org/10.1007/BFb0020275

[19] G. L. Jenkins, G. Roger, M. E. Dacey, and T. Bashford, "A rigid map neural network for anatomical joint constraint modelling," in 18th International UKSim Conference on Computer Modelling and Simulation, Cambridge, UK, April 2016, in Press.

[20] F. S. Grassia, "Practical parameterization of rotations using the exponential map," J. Graph. Tools, vol. 3, no. 3, pp. 29–48, Mar. 1998. [Online]. Available: http://dx.doi.org/10.1080/10867651.1998.10487493

[21] G. Jenkins, G. Roger, S. Macken, M. Dacey, and T. Bashford, "Learned anatomical joint constraint models with rigid map networks," International Journal of Simulation, Systems, Science and Technology, vol. 17, no. 35, pp. 31.1–31.7, August 2016, http://ijssst.info/Vol-17/No-35/paper31.pdf. [Online]. Available: http://ijssst.info/Vol-17/No-35/paper31.pdf

[22] A. Watt and M. Watt, Parameterisation of Rotation. New York: ACM Press, 1992, pp. 356–368.

[23] A. E. Engin and S. M. Chen, "A statistical database for hte biomechanical properties of the human shoulder complex 1: the kinematics of the shoulder," J Biomech, vol. 108, pp. 215–222, 1986, this may be simular to the work of Haung.

[24] X. Wang, M. Maurin, F. Mazet, N. D. C. Maia, K. Voinot, J. P. Verriest, and M. Fayet, "Three-dimensional modelling of the motion range of axial rotation of the upper arm," Journal of Biomechanics, vol. 31, no. 10, pp. 899 – 908, 1998. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021929098000980

[25] J. U. Korein, A geometric investigation of reach. Massachusetts: MIT Press, 1984.

[26] J. Wilhelms and A. V. Gelder, "Fast and easy reach-cone joint limits," J. Graph. Tools, vol. 6, no. 2, pp. 27–41, Sep. 2002. [Online]. Available: http://dx.doi.org/10.1080/10867651.2001.10487539

[27] W. Manurel and D. Thalmann, "Human shoulder joint modelling including scapulo-thoratic constraints and joint sinus cones," Compt Graph, vol. 24, pp. 203–218, 2000.

[28] M. P. Engell-Nørreg°ard, S. M. Niebe, and K. Erleben, "Local jointlimits using distance field cones in euler angle space," in Computer Graphics International, Singapore, June 2010.

[29] S. N. Engell-Nørreg°ard and K. Erleben, "A joint constraint model for human joints using signed distance-fields," Multibody Syst Dyn, vol. 28, pp. 69–81, 2012.

[30] I. Akhter and M. J. Black, "Pose-conditioned joint angle limits for 3d human pose reconstruction," in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015, pp. 1446–1455.

[31] J. Lee, "A hierarchical approach to motion analysis and synthesis for articulated figures," PhD, Korean Advanced Institute of Science and Technology, 2000.

[32] Q. Liu and C. Prakash, E, "The parameterization of joint rotation with the unit quaternion," in 7th Digital Image Computing: Techniques and Applications, C. Sun, H. Talbot, S. Ourselin, and T. Adriannsen, Eds., Sydney, Australia, 2003, pp. 410–417.

[33] L. Herda, R. Urtasun, P. Fua, and A. Hanson, "Automatic determination of shoulder joint limits using quaternion field boundaries," Int J. Rob Res, vol. 22, no. 6, pp. 419–444, 2003.

[34] M. P. Johnson, "Exploiting quaternions to support expressive interactive character motion," PhD, Massachusetts Institute of Technology, 2003.

[35] E. Brau and H. Jiang, "A bayesian part-based approach to 3d human pose and camera estimation," in 2016 23rd International Conference on Pattern Recognition (ICPR), Dec 2016, pp. 1762–1767.

[36] K. Mehrotra, C. K. Mohan, and S. Ranka, Introduction. Massachusetts: MIT Press, 1997, pp. 1–40.

[37] G. L. Jenkins and M. E. Dacey, "Evolved topology generalized multilayer perceptron (gmlp) for anatomical joint constraint modelling," in 14th International UKSim Conference on Computer Modelling and Simulation. Cambridge, UK: IEEE Computer Society, 2012, pp.107–112.

[38] G. Jenkins, "Support vector machines for anatomical joint constraint," in 11th International UKSim Conference on Computer Modelling and Simulation. Cambridge, UK: IEEE Computer Society, 2009, pp. 186–190.

[39] K. Mehrotra, C. K. Mohan, and S. Ranka, Unsupervised Learning. Massachusetts: MIT Press, 1997, pp. 155–216.

[40] G. L. Jenkins and M. E. Dacey, "Self organising maps for anatomical joint constraint," in 13th International UKSim Conference on Computer Modelling and Simulation. Cambridge, UK: IEEE Computer Society, 2011, pp. 42–47.

[41] ——, "A unit quaternion based som for anatomical joint constraint modelling," in 16th International UKSim Conference on Computer Modelling and Simulation. Cambridge, UK: IEEE Computer Society, March 2014, pp. 89–95.

[42] S. Winkler, "Model-based pose estimation of 3-D objects from camera images using neural networks," Master's thesis, University of Technology Vienna, Austria, 1996.

[43] G. S. Fishman, Monte Carlo: Concepts, Algorithms, and Applications, New York: Springer-Verlag, 1996, ch. Generating Samples, pp. 145–254.

[44] J. G¨oppert and W. Rosenstiel, "Topology-preserving interpolation in self-organizing maps," in NeuroNimes, EC2, N˜ımes, France, October 1993, pp. 425–434.

[45] Y. Cheneval, P. Demartines, and L. Tettoni, "Function approximation using an architecture based on Kohonen self-organizing maps," in Journ´ees Neurosciences et Sciences de l'Ing´enieur, Ol´eron, May 1992.